How to Make E-cash with Non-repudiation and Anonymity

Song, Ronggong; Korba, Larry

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC · CNRC

How to Make E-cash with Non-repudiation and Anonymity *

Song, R., and Korba, L.
April 2004

* published in Proceeding of the International Conference on Information Technology (ITCC 2004). Las Vegas, Nevada, USA. April 5-7, 2004. NRC 46549.

How to Make E-cash with Non-Repudiation and Anonymity

Ronggong Song and Larry Korba
Institute for Information Technology
National Research Council of Canada
Ottawa, Ontario K1A 0R6, Canada
{Ronggong.Song, Larry.Korba}@nrc.ca

Abstract

Current e-cash systems enable anonymity services to protect users' privacy, but most of them do not provide the non-repudiation service such that many problems exist in the systems like denying, losing, misusing, stealing, and double-spending, etc. This paper proposes an e-cash system in which a one-time public key is embedded in the partial blind signature to provide the non-repudiation service against the above attacks. The article also demonstrates that the combination of the partial blind digital signature and anonymous digital signature makes the e-cash systems more robust and fair than before.

1. Introduction

Internet is designed to allow computers to easily interconnect and to assure that network connections will be maintained even when various links may be damaged. But this versatility also makes it easy to compromise data security and privacy. In order to provide security and privacy protection for e-commerce applications, Chaum [1] proposed a blind signature scheme in 1982. The blind signature scheme not only retains the properties of traditional digital signatures but also supports the properties: (1) the message content is blind to the signer; (2) the message may not be traced by the signer after the signature is revealed.

These properties can be used for many e-commerce applications, e.g. electronic cash (e-cash) systems [1, 2, 3, 4, 6, 9]. One feature of e-cash is that it is easily duplicated. This makes it is necessary for the bank to implement double-spending checking. However, the double-spending checking does not provide a non-repudiation service, i.e. the bank cannot prove whether the e-cash is spent by the real owner or just a thief since the non-repudiation service needs the customer's signature which may expose the customer's identity.

In order to provide strong privacy and non-repudiation protection for the customers and build a fair e-cash system, we propose a new e-cash system using a modified partial blind signature scheme proposed by Abe [5]. In the new system, the customers first need to buy the e-cash from their bank. When the customers want to use the e-cash for online shopping through Internet later, they could use the e-cash for the payment. In the modified partial blind signature scheme, we embed a temporary anonymous public key into the blind message, which does not contain any information about the customer. Since only the owner of the e-cash has the private key corresponding to the temporary anonymous public key, the new e-cash system provides a non-repudiation service with the anonymous signature of the owner of the e-cash, i.e. if the customer really spent the e-cash before, he cannot deny the action because the bank has the signature to prove the own of the e-cash has spent it but the bank still does not know who the customer is. In addition, except for the strong privacy protection, the customer can get another benefit from the new protocol — no other person but the owner can prove that they are the owner of the e-cash even if other person has a copy of the e-cash. This makes the e-cash safer than before.

The rest of the paper is organized as follows. Abe and Fujisaki's partial blind signature protocol is briefly reviewed in the next section. In Section 3, the new e-cash architecture and protocols are proposed. In Section 4, the characteristics of the new system are described. In Section 5, the privacy and security of the new protocols are analyzed. Finally, concluding remarks are given in Section 6.

2. Review of Abe and Fujisaki's Protocol

2.1. Terminology and Notations

Terminology and notations used in the paper are defined as follows.

- *A*: a customer

- B: a bank
- ES: an e-commerce store
- ID_A: customer A's identity
- $H()$: one-way hash function
- Z_n : the integers modulo n
- Z_n^* : the multiplicative group of Z_n
- $M \bmod n$: residue of M divided by n
- $Time_A$: time stamp made by customer A
- $Sign_A$: customer A's signature
- $gcd(m, n)$: greatest common divisor of m and n
- $A \rightarrow B{:}M$: customer A sends message M to the bank B
- RM: remainder money after A purchases the e-goods
- EMD: e-goods message digest

2.2. Abe and Fujisaki's Partial Blind Signature

Abe and Fujisaki's partial blind signature scheme is designed to protect the bank's database from growing without limits since the bank needs to store all spent e-cash in its database for double-spending checking. In the scheme, each e-cash document issued by the bank contains an expiration date such that all expired e-cash recorded in the bank's database can be removed. The partial blind signature scheme is described as follows.

(1) Initializing

Based on RSA public key cryptosystem [7], the bank randomly chooses two large prime numbers p and q, and computes $n = p \cdot q$ and $\phi(n) = (p-1)(q-1)$. It then determines a pair of public and private keys (e, d), satisfying $e \cdot d \equiv 1 \pmod{\phi(n)}$ with $gcd(e, \phi(n)) = 1$, and both e and d less than $\phi(n)$. The bank publishes (e, n) and a one-way hash function H, and keeps (d, p, q) secret. Let every e-cash issued by the bank worth w dollars.

(2) Withdrawing

If a customer decides to withdraw e-cash from the bank, he/she randomly chooses two integers m and r in Z_n^*, and computes $\alpha \equiv (r^{ev}H(m) \bmod n)$ where v is a message predefined by the bank and contains an expiration date of the e-cash. The customer then sends α and v to the bank. After receiving (α, v), the bank first verifies whether or not v is correct. If it is correct, the bank sends $\beta \equiv (\alpha^{(ev)^{-1}} \bmod n)$ to the customer and deducts w dollars from the customer account in the bank.

(3) Unblinding

After receiving β, the customer computes $s \equiv (r^{-1}\beta \bmod n)$ and gets his/her e-cash (m, s, v).

(4) Depositing

When the customer uses the e-cash, the payee first verifies whether or not both v is correct and $s^{ev} \equiv H(m) \bmod n$. If they are correct, he/she then calls the bank to check whether the e-cash has been already spent, i.e. double-spending checking. If the e-cash has not been spent, the payee accepts the payment and deposits the e-cash into his/her account, and the bank stores (m, s, v) in its database for double-spending checking, and adds w dollars to the payee's account.

3. A New E-cash System

3.1. Architecture

The new e-cash system consists of several components: bank, merchant, customer, and certificate authority (CA). In the new system, the bank, merchant, and customer first need to apply and get their certificates from CA. Then, all secure communications between them can be established by Transport Layer Security channel (SSL or TLS [8, 10]) through Internet. Figure 1 depicts the new system architecture.

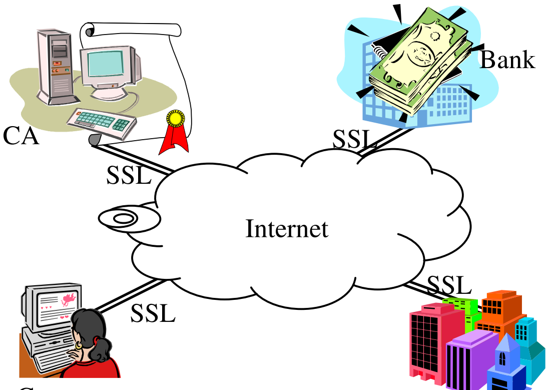

Figure 1. The new e-cash system architecture

3.2. Protocols

Based on the above partial blind signature scheme and the new e-cash system architecture, the new e-cash scheme consists of several protocols as follows.

(1) E-cash Issue protocol

In the e-cash issue protocol, we modify the above partial blind signature scheme and embed a temporary anonymous public key into the blind message such that it better suits e-cash systems and supports the non-repudiation service. When a customer wants to do online shopping, he/she first needs to buy some e-cashes issued by the bank using the following protocol where all communications are supported by the SSL security channel.

1. $A \rightarrow B$: ID_A, $Account_A$, PK_A, α, v, $Time_A$, $Sign_A$
2. $B \rightarrow A$: ID_A, ID_B, β, $Time_B$, $Sign_B$

In the above protocol, based on RSA public key cryptosystem, assume that the public and private key of the bank are (e_b , n_b) and (d_b , p_b , q_b), and the public and private key of the customer are (e_A , n_A) and (d_A , p_A , q_A), respectively. The protocol is described as follows.

Step 1: If a customer decides to buy an e-cash from the bank, he/she first makes a temporary public key (e_t, n_t), and keeps its private key (d_t, p_t, q_t) secret (using RSA public key cryptosystem). The customer then chooses a random integer r in $Z^*_{n_b}$, and computes $\alpha \equiv (r^{e_b v} H(e_t \| n_t) \ mod \ n_b)$ where || denotes the concatenation symbol, and v contains the following basic information predefined by the bank, i.e. expiration date and money.

dd/mm/yyyy	(Expiration date)
$xxx.xx	(How much money)

The customer then computes the signature $Sign_A$ as follows.

$$Sign_A \equiv (H(ID_A, Account_A, PK_A, \alpha, v, Time_A))^{d_A} \ mod \ n_A .$$

Finally, the customer uses the SSL security channel to send the messages (ID_A, $Account_A$, PK_A, α, v, $Time_A$, $Sign_A$) to the bank.

Step 2: After receiving the above messages through the SSL security channel, the bank verifies whether or not the messages: $Account_A$, $Time_A$, $Sign_A$, and v are correct. If they are correct, the bank computes $\beta \equiv (\alpha^{(e_b v)^{-1}} \ mod \ n_b)$ and the signature:

$$Sign_B \equiv (H(ID_A, ID_B, \beta, Time_B))^{d_b} \ mod \ n_b .$$

It then uses the SSL security channel to send the messages (ID_A, ID_B, β, $Time_B$, $Sign_B$) to the customer. In the meantime the bank deducts the money from the customer's account.

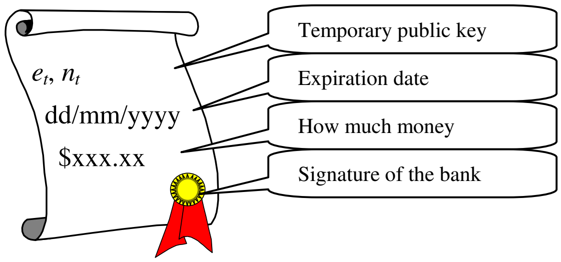

Figure 2. The digital e-cash

Finally, after receiving the messages sent by the bank through the SSL security channel, the customer verifies whether or not the messages: $Time_B$ and $Sign_B$ are correct. If they are correct, he/she then computes $s \equiv (r^{-1}\beta \ mod \ n_b)$ as the signature of the bank and gets his/her e-cash (e_t, n_t, v, s) depicted in Figure 2.

(2) Online Shopping Protocol

When the customer wants to do online shopping for some e-goods like e-book, software, and movie, etc., since it is not necessary for the shipping service, he/she could use the following protocol and e-cash to purchase and download the licenses of the e-goods if he/she wants to hide his/her identity. In the protocol, we assume that the communications also are protected with the SSL security channels.

1. $A \rightarrow ES$: E-goods, Cost, $Account_{ES}$, e_t, n_t, v, s, $Time_A$, $Sign_t$
2. $ES \rightarrow B$: Cost, $Account_{ES}$, e_t, n_t, v, s, $Time_A$, EMD, $Sign_t$
3. $B \rightarrow ES$: $Receipt_{ES}$, e_t, n_t, v, s, RM, s', $Time_B$, $Sign_B$
4. $ES \rightarrow A$: License, $Receipt_A$, e_t, n_t, v, s, RM, s', $Time_{ES}$, $Sign_{ES}$

Step 1: If the customer wants to do online shopping for some e-goods using the e-cash, he/she first selects the e-goods, and computes the following signature $Sign_t$ with the private key corresponding to the temporary public key of the e-cash,

$$Sign_t \equiv (H(Cost, Account_{ES} , e_t, n_t, v, s, Time_A) \| H(E\text{-}goods))^{d_t} \ mod \ n_t.$$

The customer then uses the SSL security channel to send the messages ($E\text{-}goods$, $Cost$, $Account_{ES}$, e_t, n_t, v, s, $Time_A$, $Sign_t$) to the merchant.

Step 2: After receiving the above messages through the SSL security channel, the merchant verifies whether or not the messages: $Cost$, $Account_{ES}$, $Time_A$, $Sign_t$, and $s^{e_b v} \equiv (H(e_t \| n_t) \ mod \ n_b)$ are correct. If they are correct, it then computes the e-goods message digest $EMD = H(E\text{-}goods)$ and forwards the messages ($Cost$, $Account_{ES}$, e_t, n_t, v, s, $Time_A$, EMD, $Sign_t$) to the bank, which issued the e-cash, through the SSL security channel.

Step 3: The bank verifies whether or not the messages: $Account_{ES}$, $Time_A$, and $Sign_t$ are correct. If they are correct, it then deposits the money into the merchant's account and deducts the money from the e-cash. The bank then computes the remainder money RM and the signature

$$s' \equiv (H(e_t, n_t, v, s, RM))^{d_b} \quad mod \ n_b .$$

$$Sign_B \equiv (H(Receipt_{ES}, e_t, n_t, v, s, RM, s', Time_B))^{d_b} \quad mod \ n_b .$$

Finally, the bank makes a statement (receipt) for the merchant and sends the messages ($Receipt_{ES}$, e_t, n_t, v, s, RM, s', $Time_B$, $Sign_B$) to the merchant through the SSL security channel.

Step 4: The merchant verifies whether all messages are correct. If correct, it makes a receipt for the customer and computes the signature

$$Sign_{ES} \equiv (H(License, Receipt_A, e_t, n_t, v, s, RM, s', Time_{ES},))^{d_{ES}} \quad mod \ n_{ES} .$$

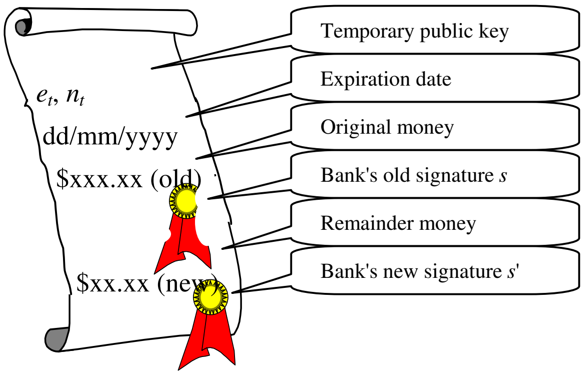

Figure 3. The remainder digital e-cash

Finally, the merchant sends the messages ($License$, $Receipt_A$, e_t, n_t, v, s, RM, s', $Time_{ES}$, $Sign_{ES}$) to the customer through the SSL security channel.

After receiving the messages, the customer gets the licenses of the e-goods and his/her remainder e-cash. Figure 3 depicts the remainder e-cash.

(3) E-cash Renew Protocol

In this protocol, the customer can renew his/her e-cash when the e-cash is close to the expiration date through the following protocol. In addition, the bank also cannot build a relationship between the old e-cash and the new e-cash through the protocol.

1. $A \rightarrow B$: $\alpha, v, e_t^{'}, n_t^{'}, v^{'}, s^{'}, Time_t, Sign_t$
2. $B \rightarrow A$: $e_t^{'}, n_t^{'}, v^{'}, s^{'}, \beta, Time_B, Sign_B$

Step 1: The customer first fills a new e-cash form and computes the new blind messages α and v as the above e-cash issue protocol, and then uses the old e-cash to compute the signature

$$Sign_t \equiv (H(\alpha, v, e_t^{'}, n_t^{'}, v^{'}, s^{'}, Time_t))^{d_t} \quad mod \ n_t.$$

Finally, the customer sends the messages (α, v, $e_t^{'}$, $n_t^{'}$, $v^{'}$, $s^{'}$, $Time_t$, $Sign_t$) to the bank through the SSL security channel.

Step 2: After receiving the messages, the bank verifies whether or not the messages are correct. If they are correct, the bank computes $\beta \equiv (\alpha^{(e_b v)^{-1}} \ mod \ n_b)$ and the signature

$$Sign_B \equiv (H(e_t^{'}, n_t^{'}, v^{'}, s^{'}, \beta, Time_B))^{d_b} \quad mod \ n_b .$$

It then records that the old e-cash is cancelled until the expiration date. After the expiration date, the bank could delete the all information about the old e-cash. Finally, the bank sends the new e-cash to the customer through the SSL security channel.

4. Protocol Characteristics

(1) Strong Privacy Protection

In the new system, anyone including the bank and merchant cannot determine to who purchases the e-goods. The bank and merchant know nothing about the customer except for how much money the customer spends for e-cashes. This provides strong privacy protection for the customers.

(2) Non-repudiation

Since all transferred messages are signed with the signatures of the owners of the messages in the new protocols, they can ask a Court to judge it if there is a dispute later, i.e. the new protocol provides the non-repudiation service. On the other hand, the signatures of the customers do not expose their private information (see detail analysis in Section 5).

(3) Strong Safety Protection

The new protocols only authorize the owner of the e-cash to use the e-cash. Other person including the bank and merchant cannot use the e-cash since they cannot make the signature without the private key of the e-cash and proof that they are the owner of the e-cash. Hence, the customers need not worry about the losing, misusing, and stealing of their e-cashes.

5. Privacy and Security Analysis

In this section, we first demonstrate that the new protocols do provide strong privacy protection for customers, and non-repudiation of acquired services, and then examine the security of the new protocols against other passive and active attacks.

5.1. Anonymity Analysis

This new protocols support the anonymity of customers through the use of partial blind signatures and anonymous temporary public key. Since the temporary public key is embedded into the blind message of the partial blind signature scheme, and the format and content of the message v are the as same as the other e-cashes, the bank and merchant cannot trace the identity information of the owner of the e-cash when the customer uses the e-cash later, i.e. the bank and merchant does not know who purchases the e-goods using the e-cash. This provides an unlinkability property inherent to a (partial) blind signature protocol.

In addition, since the e-cash is unlinkable with the owner identity, the bank would know nothing about the customer except how much money the customer exchanges for the e-cash. On the other hand, since the merchant only would have the record message about the e-cash, it also would know no more about its customers, as would any outsider. Hence, it gives the customers strong privacy protection.

5.2. Non-repudiation Analysis

The new protocols provide non-repudiation services in each step of the protocols with the signatures. First, in the e-cash issue protocol, the message that the customer sends to the bank is signed with the customer's certificate. If the customer denies this action, the bank can show the customer's signature to the Court. On the other hand, if the customer does not do this, the bank also cannot charge the customer since it cannot give an evidence (i.e., signature) to prove it.

Secondly, in the online shopping protocol, the messages sent to the merchant also are signed with the private key of the e-cash. Since only the owner of the e-cash has the private key, the owner cannot deny his/her action if he/she signed the message. On the other hand, this also makes the e-cash safer since other person cannot spend the e-cash without the private key. In addition, as we mentioned in the above anonymity analysis, this signature does not expose the identity of the owner of the e-cash since the temporary public key does not include any information about the identity of the owner, and also is embedded in the blind message in the e-cash issue protocol.

5.3. Security Analysis

(1) Passive Attacks

In the new protocols, all messages sent to the intended receiver are protected with the SSL security channels. Thus, an adversary other than the intended receiver cannot determine the content of the messages just by looking at them, i.e. the outsiders know nothing about the communication contents.

On the other hand, in the e-cash issue protocol, since the temporary public key (e_t, n_t) is embedded in the blind message $\alpha \equiv (r^{e_b v} H(e_t \| n_t) \; mod \;\; n_b)$, the bank also does not know r and $H(e_t \| n_t)$, i.e. the bank cannot readily determine who holds the temporary public key.

(2) Active Attacks

The new protocols also provide protection against replay and modification attacks. Using the time stamp "*Time*" in each message, the receiver can easily discover a replayed message. Additionally, if some adversaries want to change the messages or impersonate the customer/bank/merchant, the intended receiver can easily find out by verifying the signature "*Sign*" since all messages sent to the receiver have been hashed, and the hashing value has been signed, i.e. other person cannot change or make the messages without the private key.

6. Conclusion

We have presented a new e-cash system with strong privacy and non-repudiation protection. This new system has the following advantages over traditionally e-cash system:

- Providing strong privacy protection for customers,
- Providing non-repudiation services,
- Protecting the customer, bank, and merchant against the denying, double-spending, losing, misusing, and stealing of the e-cashes,
- Could be easily implemented with XML and the SSL security channel.

ACKNOWLEDGMENT

We would like to thank all members of IIT at the NRC of Canada and the Communications Security Establishment of Canada for their support towards our Security and Privacy R&D program. We are also grateful to the anonymous referees for helpful comments.

References

[1] D. Chaum. “Blind Signature for Untraceable Payments”. *Advances in Cryptology – Crypto'82*, pp.199-203, 1983.

[2] D. Chaum, A. Fiat, and M. Naor. “Untraceable Electronic Cash”. *Advances in Cryptology – Crypto'88* (LNCS 403), pp.319-327, 1990.

[3] J. K. Liu, V. K. Wei, and S. H. Wong. Recoverable and Untraceable E-cash. EUROCON’2001, Trends in Communications, International Conference on, Volume: 1, July 2001.

[4] Jens Bo Friis. Digicash Implementation. http://bofriis.dk/security/digicash_implementation.pdf. June, 2003.

[5] M. Abe and E. Fujisaki. “How to Date Blind Signatures”. *Advances in Cryptology – ASIACRYPT'96* (LNCS 1163), pp.244-251, 1996.

[6] P. L. Yu and C. L. Lei. An User Efficient Fair E-cash Scheme with Anonymous Certificates. Electrical and Electronic Technology, 2001. TENCON. Proceedings of IEEE Region 10 International Conference on, Vol. 1, Aug 2001.

[7] R. L. Rivest, A. Shamir, and L. Adleman. A Method For Obtaining Digital Signatures and Public-key Cryptosystems. *Communications of ACM*, Vol.21, No.2, pp.120-126, Feb 1978.

[8] SSL 3.0 Specification. http://wp.netscape.com/eng/ssl3/. 1996.

[9] T. Okamoto and K. Ohta. “Universal Electronic Cash”. *Advances in Cryptology – Crypto'91* (LNCS 576), pp.324-337, 1992.

[10] The TLS Protocol Version 1.0. RFC 2246 (Proposed Standard). ftp://ftp.rfc-editor.org/in-notes/rfc2246.txt. 1999.